INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The back and forth nudging algorithm applied to a shallow water model, comparison and hybridization with the 4D-VAR

Didier Auroux

N° 6678

Octobre 2008

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6678--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

The back and forth nudging algorithm applied to a shallow water model, comparison and hybridization with the 4D-VAR

Didier Auroux*†

Thème NUM — Systèmes numériques
Équipe-Projet MOISE

Rapport de recherche n° 6678 — Octobre 2008 — 25 pages

Abstract: We study in this paper a new data assimilation algorithm, called the Back and Forth Nudging (BFN). This scheme has been very recently introduced for simplicity reasons, as it does not require any linearization, or adjoint equation, or minimization process in comparison with variational schemes, but nevertheless it provides a new estimation of the initial condition at each iteration. We study its convergence properties as well as efficiency on a 2D shallow water model. All along the numerical experiments, comparisons with the standard variational algorithm (called 4D-VAR) are performed. Finally, a hybrid method is introduced, by considering a few iterations of the BFN algorithm as a preprocessing tool for the 4D-VAR algorithm. We show that the BFN algorithm is extremely powerful in the very first iterations, and also that the hybrid method can both improve notably the quality of the identified initial condition by the 4D-VAR scheme and reduce the number of iterations needed to achieve convergence.

Key-words: Data assimilation, shallow water, back and forth nudging, variational methods, hybrid method, nonlinear dynamics

* Institut de Mathématiques de Toulouse, Université Paul Sabatier Toulouse 3, 31062 Toulouse cedex 9, France (auroux@math.univ-toulouse.fr)
† INRIA Grenoble Rhône-Alpes, équipe-projet MOISE, France

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Le nudging direct et rétrograde appliqué à un modèle shallow-water: comparaison et hybridation avec le 4D-VAR

Résumé : Dans cet article, nous étudions le nudging direct et rétrograde. Cet algorithme a été récemment introduit pour des raisons de simplicité, comme il ne nécessite ni linéarisation, ni état adjoint, ni minimisation, par rapport aux méthodes variationnelles. Cet algorithme fournit toutefois une nouvelle estimation de la condition initiale à chaque itération. Nous étudions les propriétés de convergence et l'efficacité de cet algorithme sur un modèle shallow-water 2D. Tout au long des expériences numériques, nous effectuons des comparaisons avec le 4D-VAR. Enfin, une méthode hybride est introduite dans cet article, consistant à effectuer quelques itérations de l'algorithme BFN comme préconditionnement de l'algorithme 4D-VAR. Nous montrons que l'algorithme BFN est très performant dans les toutes premières itérations, et la méthode hybride peut non seulement améliorer sensiblement la qualité de la condition initiale identifiée par le 4D-VAR, mais aussi réduire la nombre d'itérations nécessaires pour atteindre la convergence.

Mots-clés : Assimilation de données, modèle shallow-water, nudging direct et rétrograde, méthodes variationnelles, hybridation, dynamique non linéaire

1 INTRODUCTION

Data assimilation consists of estimating the state of a system by combining via numerical methods two different sources of information: models, and observations. Data assimilation makes it possible to answer a wide range of questions such as: the optimal identification of the initial state of a system, and then reliable numerical forecasts; the systematic identification of error sources in the models; the optimization of observation networks; the extrapolation, using a numerical model, of the values of non observed variables. Thus, data assimilation is increasingly used in the community of geophysical sciences, in relation with mathematicians. There are two main classes of data assimilation methods, the first based on estimation theory (sequential assimilation), and the other based on optimal control theory (variational assimilation) [6, 12].

The most sophisticated variational method, currently used in many centers of operational forecast (in oceanography and in meteorology), is the 4D-VAR (four-dimensional variational) algorithm [13]. It consists of assimilating all the available information (contained both in the model and the observations) during the work (or assimilation) period. The problem of identifying the state of a system at a given time can then be written as the minimization of a criterion measuring the difference between the forecasts of the model and the observations of the system in the given time window. In general, the initial state of the time interval is taken as the control variable for the minimization process. This provides an advantage to the 4D-VAR algorithm is that for each time step, the estimation of the state vector depends not only on the previous observations, but also on the future observations. Propagative phenomena, such as waves, are generally well represented by the 4D-VAR method. The disadvantages of the 4D-VAR algorithm are on one hand its quite difficult implementation, because it requires both the adjoint of the physical model and a powerful minimization algorithm, and on the other hand the lack of estimation of the errors in the assimilated state. Contrarily to the 4D-VAR, the sequential methods only require the physical model in the direct mode.

The spearhead of sequential methods, which are also operational (but more marginally than 4D-VAR), is the Kalman filter. The Kalman filter is designed to provide, for each time step, the optimal estimate (of variance of minimal error) of the system state, by using only the estimates of the state and the last observations. It alternates propagation steps (with the physical model) and correction steps (using the observations) of the state and of its error statistics. The main advantage of the Kalman filter is that it provides *in real time* an estimation of the statistics of errors of the state, in addition to the state itself, and thus it is able to provide a statistically optimal estimate of the state. Its weakness is its inability to take into account future observations like the 4D-VAR algorithm does. Extended forms of the Kalman filter are designed to integrate future observations then to smooth the model trajectory [17, 9]; they are called the Kalman smoothers. 4D-VAR and Kalman filter or smoothers can be shown to be theoretically equivalent under certain hypotheses. However, the assumptions necessary to implement them are usually different and the equivalence is always lost in practice.

Nudging can be seen as a degenerate, oversimplified form of the Kalman filter. Sometimes called *the poor man's assimilation method*, it consists of applying a Newtonian recall of the state value towards its direct observation. In

spite of the differences, the term *nudging* is also sometimes used in the context of statistical interpolation. The standard nudging algorithm, which initially appeared in meteorology [11], is the first data assimilation method used in an operational way in oceanography [15, 16, 14]. Some recent studies have shown that it is possible to determine in a systematic way the optimal weighting coefficients of the recall force to the observations, and then nudging is equivalent to the Kalman filter, or in a variational form, to 4D-VAR [18, 19].

One of the main disadvantages of the sequential data assimilation methods is that they cannot, at a given time, take into account the future observations. They do not improve the estimation of the initial condition of the system. A simple idea, allowing at the same time the improvement of the estimation of the initial condition and the assimilation of future observations, consists of applying a second time the sequential method, but on the backward (in time) model, using the estimation of the final state (obtained by the forward assimilation) as a new initial guess. Thus, at the end of this process, one obtains a new estimation of the system state at the initial time (which makes it possible to use this estimation in a variational data assimilation method), and at each time of the backward assimilation, the corrections actually use the previous observations in the backward model, therefore future observations, to improve the estimated states. The backward problems are generally very ill posed (the Laplace equation, or heat equation, is a very good example), and it is not a priori easy to apply a traditional data assimilation method to a backward model. However, if one uses a degraded version of the Kalman filter for complete observations of the system (in time and space), Auroux and Blum have shown that it is possible to stabilize backward integrations thanks to the assimilation corrective term [4, 5].

Auroux et al. proposed in [4] an original approach of backward and forward nudging (or *back and forth* nudging, BFN), which consists of initially solving the forward equations with a nudging term, and then using the final state thus obtained as initial condition to solve the same equations in a backward direction with a feedback term (with the opposite sign compared to the feedback term of forward nudging). This process is then repeated in an iterative way until convergence of the initial state is achieved. The basic idea of the BFN algorithm thus allows the use of a sequential data assimilation method (here, nudging) while taking into account the future observations, as in 4D-VAR. On the other hand, contrarily to 4D-VAR, the adjoint of the physical model and the minimization algorithm are not needed. The implementation of the BFN algorithm is thus vastly simplified in comparison with the 4D-VAR algorithm. This can be of great value in the present context of more and more complex models and increasingly diversified observations of the ocean or the atmosphere. Indeed many research teams cannot support the human cost for the implementation of traditional data assimilation methods such as the 4D-VAR.

In this article, we propose to study the behaviour of the BFN algorithm on a 2D shallow water model, and then to compare it with the standard 4D-VAR algorithm (or variational algorithm, as in some sense the problem is not really 4D as there are only two dimensions in space) on several points such as the observation errors, the model error, ... As the BFN algorithm provides an estimation of the initial condition at each iteration, like all variational schemes, a new idea is also studied, by considering a preprocessing of the 4D-VAR algorithm with the BFN algorithm. Such a hybrid method may combine the advantages of

the BFN scheme (high performance in the very first iterations) with the known efficiency of the 4D-VAR, without any additional fastidious implementation. We work on a shallow water model for simplicity reasons, as it is a quite light model considered for simple numerical experiments in geophysics, but it mimics quite well the evolution of geophysical flows. Moreover, it is possible to consider observations of only one model variables, for instance the fluid height, and to study the identification of all the model variables (fluid height and velocity).

This paper is organized as follows. We first detail in section 2 the shallow water model we have considered in all the numerical experiments, by giving the equations and the values of the model parameters. We also recall very briefly the BFN algorithm, and we present its application to the shallow water model equations. Then, we report in section 3 the results of extensive numerical simulations with synthetic data. We first study the convergence of the BFN scheme, and its comparison with the standard variational algorithm in the case of perfect observations, and then with noisy observations. We introduce then the new hybrid BFN-4DVAR method, by considering a preprocessing of the 4D-VAR algorithm with a few BFN iterations in order to improve significantly and very rapidly the quality of the initial condition. Then, we study the sensitivity of these two schemes, as well as the hybrid method, with respect to the spatial and temporal distribution of the observations, and we also study their efficiency to assimilate data generated with a different model from the one used for the assimilation. Finally, a few concluding remarks and perspectives are given in section 4.

2 SHALLOW WATER MODEL AND BACK AND FORTH NUDGING ALGORITHM

2.1 Description of the model

The shallow water model (or Saint-Venant's equations) is a basic model, representing quite well the temporal evolution of geophysical flows. This model is usually considered for simple numerical experiments in oceanography, meteorology or hydrology. The shallow water equations are a set of three equations, describing the evolution of a two-dimensional horizontal flow. These equations are derived from a vertical integration of the three-dimensional fields, assuming the hydrostatic approximation, i.e. neglecting the vertical acceleration. There are several ways to write the shallow water equations, considering either the geopotential or height or pressure variables. We consider here the following configuration:

$$\left\{\begin{array}{l} \partial_t u - (f+\zeta)v + \partial_x B = \dfrac{\tau}{\rho_0 h} - ru + \nu\Delta u, \\ \partial_t v + (f+\zeta)u + \partial_y B = \dfrac{\tau}{\rho_0 h} - rv + \nu\Delta v, \\ \partial_t h + \partial_x(hu) + \partial_y(hv) = 0, \end{array}\right. \tag{1}$$

where the unknowns are u and v the horizontal components of the velocity, and h the geopotential height (see e.g. [1, 7]). The initial condition $(u(0), v(0), h(0))$ and no-slip lateral boundary conditions complete the system. The other parameters are the following:

- $\zeta = \partial_x v - \partial_y u$ is the relative vorticity;
- $B = g^* h + \frac{1}{2}(u^2 + v^2)$ is the Bernoulli potential;
- $g^* = 0.02\ m.s^{-2}$ is the reduced gravity;
- $f = f_0 + \beta y$ is the Coriolis parameter (in the β-plane approximation), with $f_0 = 7.10^{-5}\ s^{-1}$ and $\beta = 2.10^{-11}\ m^{-1}.s^{-1}$;
- $\tau = (\tau_x, \tau_y)$ is the forcing term of the model (e.g. the wind stress), with a maximum amplitude of $\tau_0 = 0.05\ s^{-2}$;
- $\rho_0 = 10^3\ kg.m^{-3}$ is the water density;
- $r = 9.10^{-8}\ s^{-1}$ is the friction coefficient.
- $\nu = 5\ m^2.s^{-1}$ is the viscosity (or dissipation) coefficient.

We consider a numerical configuration in which the domain is a square of 2000 km $\times$2000 km, with a rigid boundary, and no-slip boundary conditions. The time step is 1800 seconds (half an hour), and we consider an assimilation period $[0; T]$ of 720 time steps (i.e. 15 days). The forecast period is $[T; 4T]$, corresponding to 45 prediction days. The spatial resolution is 25 kilometers. The wind forcing is chosen constant in time, and set to a sine function which induces a standard double gyre circulation. This numerical model has been developed by the MOISE research team of INRIA Rhône-Alpes [8].

We briefly describe the numerical schemes used for the resolution of equations (1) and we refer to [8] for more details. We consider a leap-frog method for time discretization of equations (1), controlled by an Asselin time filter [3]. The equations are then discretized on an Arakawa C grid [2], with $N \times N$ points ($N = 81$ in our experiments): the velocity components u and v are defined at the center of the edges, and the height is defined at the center of the grid cells. Then, the vorticity and Bernoulli potential are computed at the nodes and center of the cells respectively.

The initial conditions are $u = v = 0$ and $h = 500$ meters. The spin-up phase lasts nearly 6 years, after which the model simulates a double-gyre wind-driven oceanic circulation. This approximate model reproduces quite well the surface circulation at mid-latitudes, including the jet stream and ocean boundary currents. In our experiments, the water depth varies from roughly 265 to 690 meters, its mean being 500 meters, and the maximum velocity (in the jet stream) is roughly $1.1\, m.s^{-1}$, the mean velocity being $0.1\, m.s^{-1}$. Figure 1 shows the height h and longitudinal velocity u at the reference state (see section 2.3).

2.2 BFN algorithm applied to this model

The back and forth nudging (BFN) algorithm, introduced in [4], consists of repeatedly performing forward and backward integrations of the model with relaxation (or nudging) terms, using opposite signs in the direct and inverse integrations, so as to make the backward evolution numerically stable. The aim of the nudging term (or feedback to the observations) is also to assimilate the data. After each iteration (consisting of one forward and one backward numerical integrations), one obtains an estimation of the initial condition of the

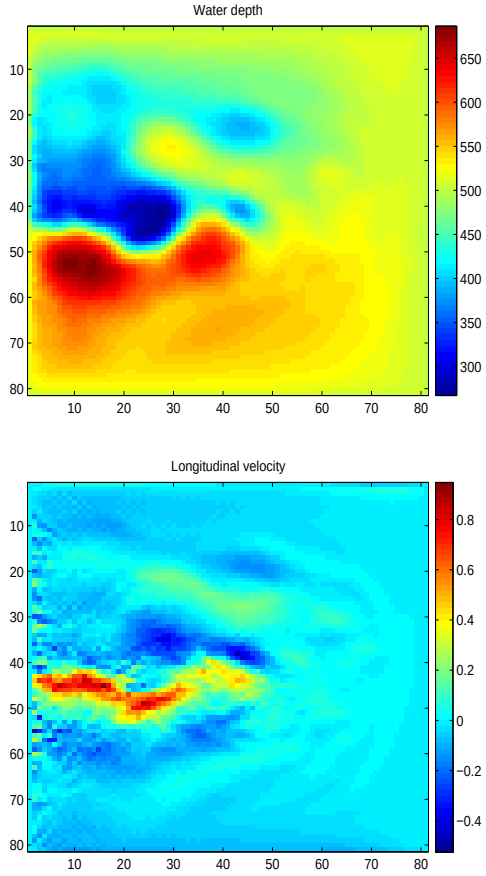

Figure 1: Height h (in meters) and longitudinal velocity u (in $m.s^{-1}$) of the ocean at a reference state.

system. We repeat these forward and backward integrations (with the feedback terms) until convergence of the algorithm. We refer to [4] and [5] for more details about this algorithm.

The application of the BFN algorithm to the previous shallow water model equations leads to the integration of the following set of equations, for $k \geq 1$:

$$\begin{cases} \partial_t u^k - (f+\zeta^k)v^k + \partial_x B^k = \dfrac{\tau}{\rho_0 h^k} - ru^k + \nu\Delta u^k, \\ \partial_t v^k + (f+\zeta^k)u^k + \partial_y B^k = \dfrac{\tau}{\rho_0 h^k} - rv^k + \nu\Delta v^k, \qquad 0 < t < T, \\ \partial_t h^k + \partial_x(h^k u^k) + \partial_y(h^k v^k) = K(h_{obs} - Hh^k)\delta_{obs}, \\ (u^k(0), v^k(0), h^k(0)) = (\tilde{u}^{k-1}(0), \tilde{v}^{k-1}(0), \tilde{h}^{k-1}(0)), \end{cases} \tag{2}$$

$$\begin{cases} \partial_t \tilde{u}^k - (f+\tilde{\zeta}^k)\tilde{v}^k + \partial_x \tilde{B}^k = \dfrac{\tau}{\rho_0 \tilde{h}^k} - r\tilde{u}^k + \nu\Delta\tilde{u}^k, \\ \partial_t \tilde{v}^k + (f+\tilde{\zeta}^k)\tilde{u}^k + \partial_y \tilde{B}^k = \dfrac{\tau}{\rho_0 \tilde{h}^k} - r\tilde{v}^k + \nu\Delta\tilde{v}^k, \qquad T > t > 0, \\ \partial_t \tilde{h}^k + \partial_x(\tilde{h}^k\tilde{u}^k) + \partial_y(\tilde{h}^k\tilde{v}^k) = -K'(h_{obs} - H\tilde{h}^k)\delta_{obs}, \\ (\tilde{u}^k(T), \tilde{v}^k(T), \tilde{h}^k(T)) = (u^k(T), v^k(T), h^k(T)), \end{cases} \tag{3}$$

with the initialization $(\tilde{u}^0(0), \tilde{v}^0(0), \tilde{h}^0(0)) = (u_b, v_b, h_b)$, where we denote by (u_b, v_b, h_b) the background estimation of the initial condition. The Bernoulli potentials B^k and $\tilde{B}^k$, and the vorticities ζ^k and $\tilde{\zeta}^k$ are defined as in the standard direct model (see previous subsection). Finally, H is the observation operator, and δ_{obs} is a time Dirac function, corresponding to the observation time steps: when there are no available observations at a given time, then there is no nudging correction.

Equations (2) are discretized by the same numerical scheme as the one used for equations (1) (see previous subsection). The only difference between these two sets of equations is indeed the feedback term in the height equation. Thus, we need to assume that the observations h_{obs} are available at the same locations as the height h^k, which is the case here, as we consider twin experiments (see next subsection). Finally, the backward equations (3) are also discretized by the same schemes.

As we assume that there are no observations of the velocity field, we do not consider any correction terms in their corresponding equations. However, from the algorithmic point of view, it is possible to correct the velocity with the $h_{obs} - Hh$ term, or a linear function of this term (for instance, its partial derivative). In the following, only the height equations are corrected.

As explained for instance in [5], we will consider the following nudging matrices, for simplicity reasons:

$$K = kH^T, \qquad K' = k'H^T, \tag{4}$$

where k and k' are real positive scalars, and H^T is the adjoint of the observation operator. This choice is justified by the fact that H^T allows the corrections provided by the observations to be applied exactly at the observation locations. Moreover, in the case of noisy observations, we will assume that the observation errors are spatially uncorrelated and white gaussian distributed, with a constant

variance. In these conditions, the covariance matrix of observation errors is simply proportional to the identity matrix (of the observation space), and this also justifies equation (4).

2.3 Experimental approach for the numerical studies and comparisons with a 4D-VAR algorithm

In all the following numerical experiments, we will consider twin experiments: a reference initial condition is set, and some data are extracted from the corresponding trajectory. These data are then noised for some experiments (and not for some others), and provided as observations to the BFN and 4D-VAR data assimilation algorithms.

We assume that some observations h_{obs} of only the height h are available, every n_t time steps and every n_x grid points (in both longitudinal and transversal directions). We can then easily define an observation operator H, from the model space to the observation space, as an extraction of one every n_x values of the variable h in each direction. This operator is clearly linear, and allows us to compare a model solution with the observations. Unless some other values are given, the considered values of n_x and n_t are respectively 5 and 24.

In such a configuration, the model space (state variables (u, v, h)) is of dimension 19683, and the observation space (data variable h_{obs}) is of dimension 289. The corresponding total number of observations all over the assimilation period is 8959.

We will study different points, the first one being the convergence of the BFN scheme. For this purpose, we will simply study the stabilization of the BFN trajectories with the iterations.

Then, for both BFN and 4D-VAR algorithms, we will study the error on the identification of the initial condition. As we only work with simulated data (extracted from a reference trajectory), it is easy to compare the identified solutions with the true state, by considering the relative error:

$$\frac{\|h^k(0) - h_{true}(0)\|}{\|h_{true}(0)\|}, \tag{5}$$

where $\|\,.\,\|$ represents here the standard discrete L^2 norm in space, and $h^k(0)$ represents the initial height identified by either the BFN or the 4D-VAR schemes after k iterations. By considering the forecast trajectories corresponding to this initial condition, we will also study the relative difference at time t between the forecast solution associated to the identified initial condition and the true forecast solution. We will also consider similar quantities for the velocity u and v.

We now give some details about the 4D-VAR algorithm we have used for the numerical comparisons [13]. We consider an incremental 4D-VAR, in which the control vector is the increment to the background state:

$$J(\delta u_0, \delta v_0, \delta h_0) = \frac{1}{2}\|(\delta u_0, \delta v_0, \delta h_0)\|^2_{B^{-1}} + \frac{1}{2}\sum_i \|h_{obs}(t_i) - Hh(t_i)\|^2_{R^{-1}}, \tag{6}$$

where (t_i) represent the observation times, and B and R are the covariance matrices of background and observation errors respectively. The adjoint state

provides the gradient of this cost function, and the minimization is performed by a BFGS quasi-Newton algorithm [10].

For the comparisons between the 4D-VAR and BFN schemes, we will consider either a similar number of iterations, or consider the solutions at convergence. The first approach is justified by a comparable numerical cost for both algorithms, as one BFN iteration consists of one forward and one backward integration of the model equations (with a nudging term), and one 4D-VAR iteration also consists of one forward integration (direct model equation) and one backward integration (adjoint equation). This will give an idea about the efficiency of the schemes, for a given computational cost. The second approach will on the contrary provide the efficiency of the schemes when convergence is achieved, but also the corresponding time needed to reach convergence.

The background initial condition, used for the initialization of both the 4D-VAR and BFN schemes, is the true state of the ocean but two weeks before, to which we added a small bias and a white gaussian noise of small relative variance. The covariance matrix B of observation errors is set accordingly to the geostrophic projection [7]:

$$B^{-\frac{1}{2}}(u_0; v_0; h_0)^T = \left(u_0 - \frac{g}{f}\frac{\partial h_0}{\partial y}; v_0 + \frac{g}{f}\frac{\partial h_0}{\partial x}; h_0\right)^T. \tag{7}$$

We recall that R is simply set proportional to the identity matrix, as the observations are independently noised.

Finally, the backward nudging coefficient k' is usually chosen to be the smallest coefficient that makes the numerical backward integration stable [5]. In the standard configuration of observation density ($n_x = 5$, $n_t = 24$), the minimal value is of the order of $10^{-6}\, s^{-1}$. We have then set $k' = 10^{-5}\, s^{-1}$ in all the corresponding numerical experiments. In the case of sparse observations, k' must be increased in order to stabilize the backward integrations. For instance, in the sparsest case ($n_x = 20$, $n_t = 72$), we have set $k' = 5 \times 10^{-4}\, s^{-1}$. The forward nudging coefficient k has to be non-negative in order to keep the continuous forward equations stable. However, note that the discretized model remains stable only if k is not too large. We set $k = k'$ in all the following numerical experiments.

3 NUMERICAL EXPERIMENTS

3.1 Convergence of the BFN with perfect observations

We first study the numerical convergence of the BFN algorithm on the shallow water model, with perfect observations. We recall that the observations are available every 5 gridpoints and every 24 time steps.

Figure 2 shows the relative difference between the height of the BFN iterates and the true height versus time, for the 5 first iterations. The height of the background state (used for the initialization of the algorithm) has a relative error of 37.6% with the true height at initial time. We can see that the first correction, provided by the observation at initial time, induces some oscillations in the solution. This is due to the fact that the observation operator is simply a restriction of the space dimension. Consequently, the trajectory is corrected (or perturbed) on every 5 gridpoints, and nowhere else. But after the first or

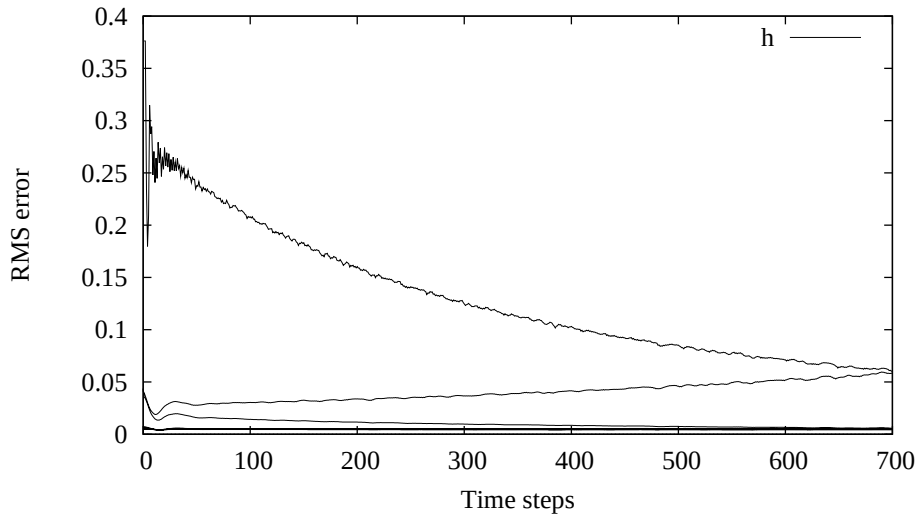

Figure 2: Relative difference between the BFN iterates (5 first iterations) and the true solution versus the time steps, for the height h.

two first corrections with the observations, there are no more oscillations. The error decreases with time during the first forward integration, and reaches 5.99% at the end of the assimilation period. Then, the backward integration is not only stabilized, but also makes the error to decrease. After one iteration (one forward and one backward integration), the relative error is equal to 4.13%. Then, the second iteration begins, and the error still decreases, both in the forward and backward numerical integrations. The fourth and fifth iterates are the same, for both the forward and backward trajectories. The relative error at initial time is 37.6% for the initialization, 4.13% after one iteration, 0.69% after two iterations, 0.51% after three iterations, 0.45% after four iterations, and 0.44% after five iterations. We can then consider that the algorithm has reached convergence, in 5 iterations.

Figure 3 shows similar results, for the two other variables: u and v. The global behaviour of the solution is the same, both versus the time steps and versus the iterations. The relative error of the background state is respectively 21.7% for u and 30.3% for v. After convergence (5 iterations), the relative error on the initial condition is 1.78% and 2.41% respectively. This is extremely noticeable, as there are no correction terms in the BFN velocity equations. Moreover, the presence of diffusion in these two equations should be a problem for the numerical backward integrations. But of course, the convergence of the height towards the true solution and the coupling between all the variables make the velocity equations stabilized and the variables u and v corrected.

These two figures show that the BFN algorithm reaches convergence is 5 iterations, and that it is an extremely efficient data assimilation method, as the error on the initial condition is divided by 6 to 10 in a very small number of iterations. We will then compare it with the 4D-VAR algorithm in various numerical experiments.

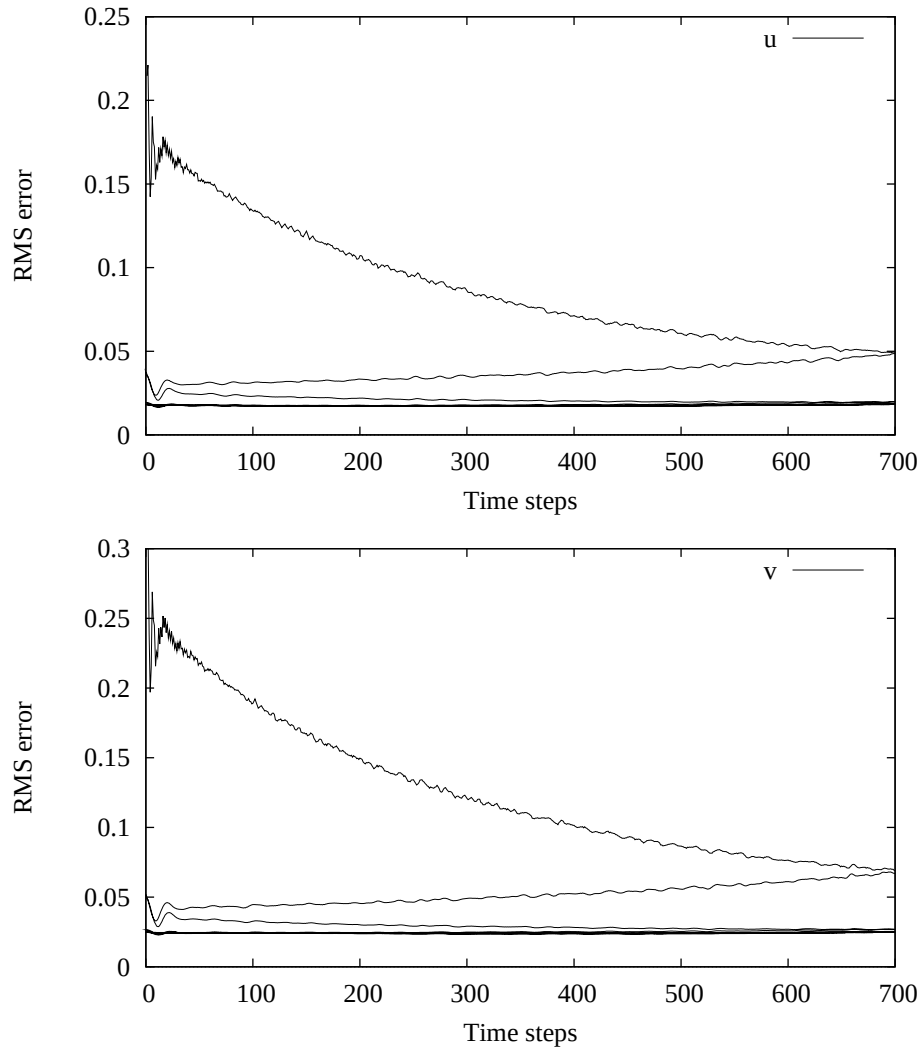

Figure 3: Relative difference between the BFN iterates (5 first iterations) and the true solution versus the time steps, for the velocity u and v.

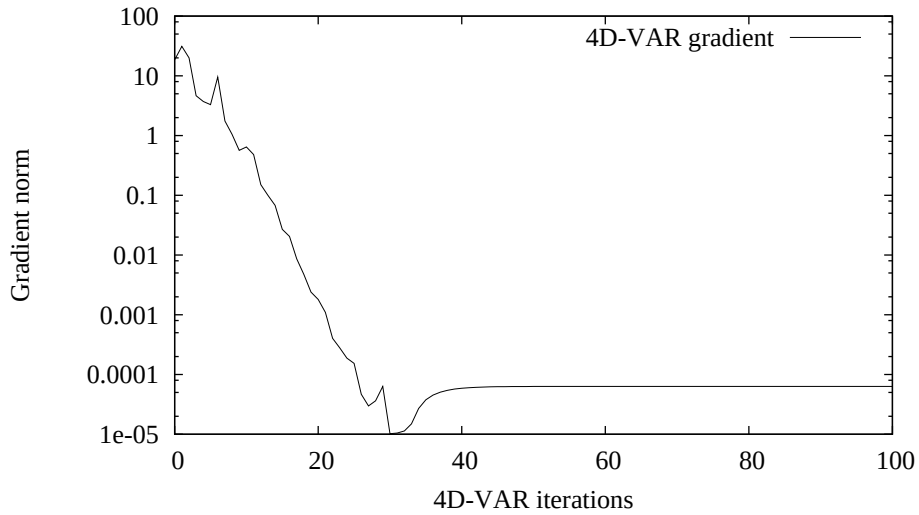

Figure 4: Norm of the gradient of the 4D-VAR cost function versus the number of iterations.

3.2 Comparison between BFN and 4D-VAR with perfect observations

In this section, we compare the BFN and 4D-VAR schemes, still with perfect observations, available every 5 gridpoints and 24 time steps. We first consider the convergence of the 4D-VAR scheme, as we have already observed the convergence of the BFN algorithm in a previous section. The initialization vector of the 4D-VAR algorithm is the background state, which means that in the incremental approach, the starting point is 0 (no increment to the background state). As the observations are perfect, we impose a large weight on the observation part of the cost function and a relatively small one on the background feedback. This choice will lead to a fair comparison, because if we consider a standard equilibrium between these two terms, as the background is not equal to the true state whereas the observations are perfect, the solution identified by the 4D-VAR would not be very close to the true state, but at some intermediate point between the true state (true observations) and the background state.

Figure 4 shows the evolution of the gradient norm of the 4D-VAR cost function during the minimization process. This figure clearly shows the convergence of the 4D-VAR algorithm. The minimization stopped after 30 iterations (in the minimization process), and 31 simulations (number of gradient computations), and then the algorithm was unable to find a better minimum. Moreover, in less than 30 iterations, the norm of the gradient has been divided by more than 10^5. We now consider the decrease of the gradient nom of 10^4 as a stop criterion for the convergence of the 4D-VAR. We will then study the solution identified by the 4D-VAR, after both 5 iterations (in order to have a comparable computing time with the BFN) and after 18 iterations in this case (when convergence is reached, with the 10^4 gradient decrease criterion.).

Relative error	h	u	v
Background state	37.6%	21.5%	30.3%
BFN (5 iterations, converged)	0.44%	1.78%	2.41%
4D-VAR (5 iterations)	0.64%	3.14%	4.47%
4D-VAR (18 iterations, converged)	0.61%	2.43%	3.46%

Table 1: Relative error of the background state and various identified initial conditions for the three variables.

Table 1 gives the relative error of the background state (i.e. relative difference with the true initial condition), the BFN identified initial condition (after 5 iterations, as convergence is reached), and the initial conditions identified by the 4D-VAR (after 5 iterations, and after 18 iterations when convergence is reached). The first two lines recall the values given in the previous subsection. It is interesting to see that the solution identified by the BFN after only 5 iterations is nearly as close to the true state as the solution identified at convergence. This shows that after 5 iterations, the solution is not very far from the local minimum identified at convergence. Another point is that the initial condition identified by the BFN at convergence is better than the 4D-VAR one. This point can be explained by the error on the background state. On one hand, the BFN only works with the observations, the background begin used only for the initialization. But on the other hand, the 4D-VAR cost function to be minimized has a background term, and even if it is small compared with the observation term (as there are no observation errors), it cannot be set too small for computational reasons. In this case, the identified solution is still a compromise between the observations and the wrong background state.

Another way to study the efficiency of the scheme is to consider the forecast evolution of the identified solution, as the quality of the initial condition itself is not very important in some sense. For instance, figure 5 shows on the top the relative difference between the true solution and the forecast solution corresponding to the initial condition identified by the BFN scheme after 5 iterations. We recall that the assimilation period lasts 720 time steps (or 15 days), and the forecast period ends 2880 time steps (or 2 months) after the initial time. During the assimilation period, the error on the height remains constant. In fact, it decreases a little bit during 300 iterations, and then increases a little bit after. The error on the velocity variables has a similar behaviour: first decreasing and then increasing in time. The stability of the error is interesting, and shows that the identified solution remains close to the true state all long the assimilation period. Then, the error increases more quickly, particularly on u and v. After 2 months, the relative error is 1.06% on h, 5.22% on u and 6.88% on v. It is nearly 2 to 3 times the error on the initial condition, but the most interesting point is that it is still many times smaller than the background error. The fact that the error does not increase too much with time shows that it will be possible to use this forecast solution as a good background state for data assimilation on a later period.

A similar experience on the 4D-VAR identified solution is shown on the bottom of the same figure. The forecast solutions have a similar behaviour, and the error at the end of the prediction period is nearly 2.18% for h, 9.73% for u

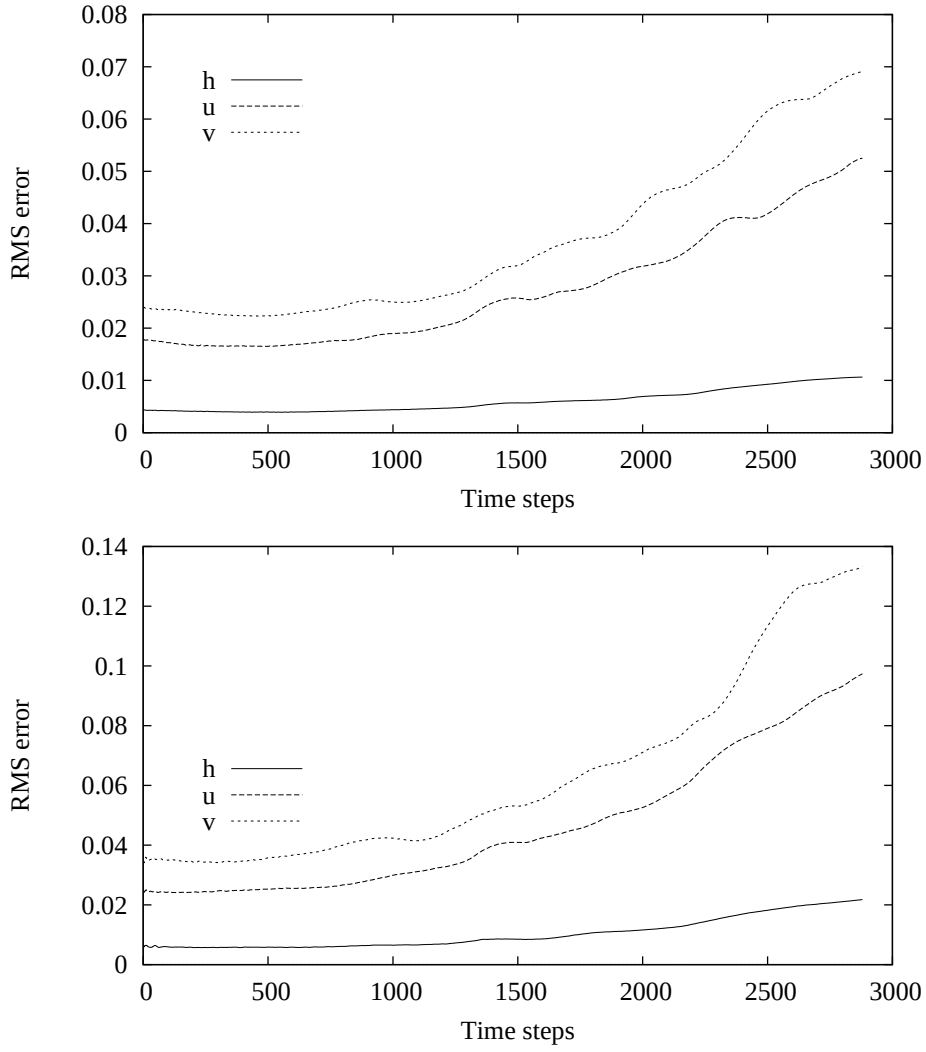

Figure 5: Relative difference between the true solution and the forecast trajectory corresponding to the BFN (top) and 4D-VAR (bottom) identified initial conditions at convergence, versus time, for the three variables.

Relative error	h	u	v
BFN (converged, $t = 0$)	3.34%	9.51%	13.6%
4D-VAR (converged, $t = 0$)	1.26%	7.12%	10.2%
4D-VAR (5 iter., $t = 0$)	1.83%	11.3%	16.4%
BFN (converged, $t = T$)	2.15%	6.74%	10.1%
4D-VAR (converged, $t = T$)	1.93%	9.20%	12.8%
4D-VAR (5 iter., $t = T$)	2.68%	13.5%	18.9%
BFN (converged, $t = 4T$)	3.67%	17.0%	24.1%
4D-VAR (converged, $t = 4T$)	4.69%	21.9%	25.9%
4D-VAR (5 iter., $t = 4T$)	5.60%	25.9%	30.9%

Table 2: Relative error of the forecast solutions corresponding to the BFN (5 iterations, converged), 4D-VAR (16 iterations, converged) and 4D-VAR (5 iterations) identified initial conditions, for the three variables, at various times: initial time, end of the assimilation period, and end of the prediction period.

and 13.3% for v. This is still much less than the background error. The error on the initial condition is nearly 1.5 times the error of the BFN state, and the error at final time is nearly twice the corresponding error of the BFN. This confirms the efficiency of the BFN algorithm. But of course, as previously explained, the efficiency of the 4D-VAR algorithm is quite degraded by the background error, and in the case of perfect observations, it is not possible to consider only the observation part of the cost function as it would lead to a very bad (an ill-posed) minimization. We will then consider now noisy observations, with a quite large level of noise, in order to compare the two algorithms in a more realistic and fair situation.

3.3 Comparison between BFN and 4D-VAR with noisy observations

The observations are now noised, with an additive white gaussian noise, with a root mean square error between 20 and 40% of the observation norm. As we know all levels of noise (on the observations and on the background state), we can set the weights of the 4D-VAR cost function to their optimal (or most realistic) values, in order to have the a priori best possible solution. From now on, we will only show figures on the forecast evolutions, as it gives both the estimation error on the initial condition, and the forecast error.

Figure 6 shows the time evolution of the forecast error corresponding to the BFN (top) and 4D-VAR (bottom) solutions when convergence is reached, in the case of noisy observations. The first point is that the quality of the results is degraded, in comparison with the case of perfect observations. Recall that the relative level of observation noise is more than 20%. The initial condition identified by the 4D-VAR algorithm is better than the BFN one, as shown in table 2. The number of iterations needed to achieve convergence is respectively 16 for the 4D-VAR and 5 for the BFN. But the very interesting thing is that at the end of the assimilation period, the BFN solution is better than the 4D-VAR one, except for the height variable, as the 4D-VAR forecast error increased nearly all the time, and the BFN one decreased also nearly all the time. The height

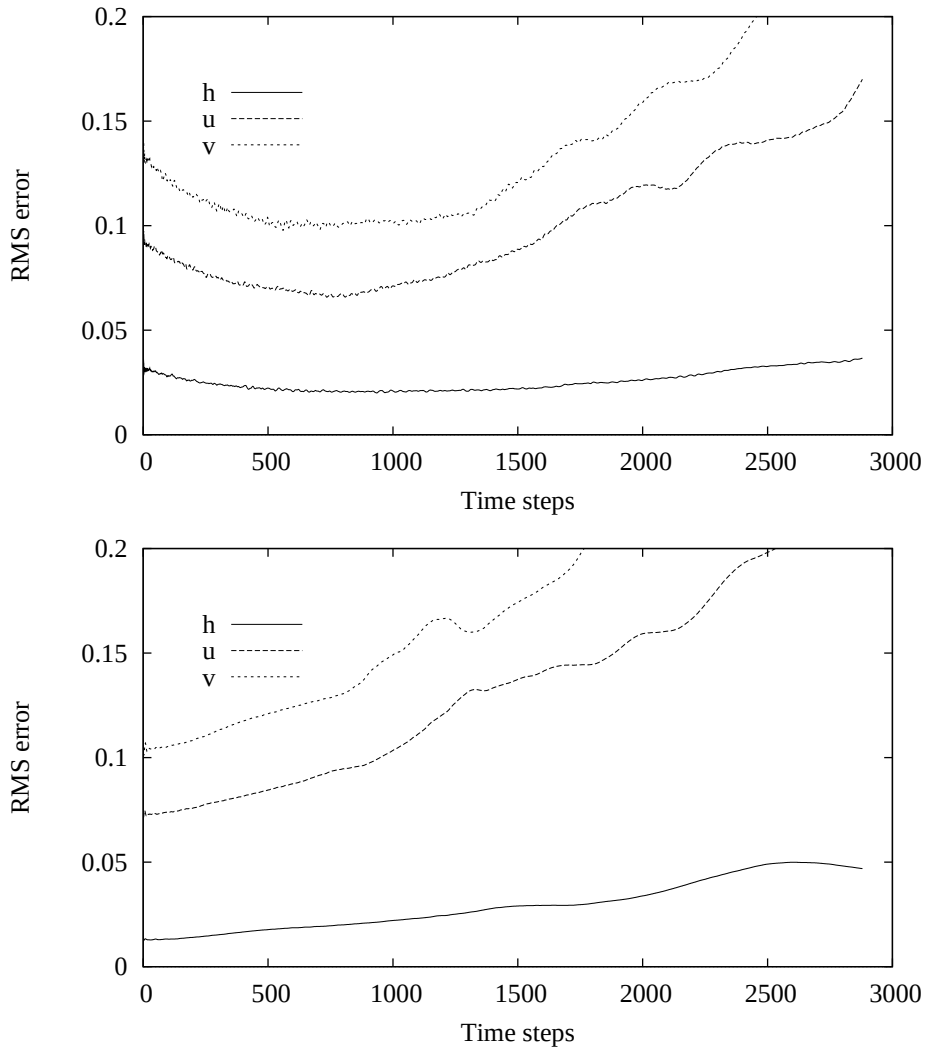

Figure 6: Relative difference between the true solution and the forecast trajectory corresponding to the BFN (top) and 4D-VAR (bottom) identified initial conditions at convergence, versus time, for the three variables and in the case of noisy observations.

Relative error	h	u	v
BFN-preproc. 4D-VAR ($2+3$ iter., $t=0$)	1.39%	7.34%	10.4%
BFN-preproc. 4D-VAR (converged, $t=0$)	1.24%	7.03%	10.4%
BFN-preproc. 4D-VAR ($2+3$ iter., $t=0$)	1.98%	9.22%	12.8%
BFN-preproc. 4D-VAR (converged, $t=0$)	1.95%	9.18%	12.9%
BFN-preproc. 4D-VAR ($2+3$ iter., $t=0$)	4.92%	22.2%	25.8%
BFN-preproc. 4D-VAR (converged, $t=0$)	4.92%	22.1%	25.6%

Table 3: Relative error of the forecast solutions corresponding to the BFN-preprocessed 4D-VAR (2+3 iterations; and 2+6 iterations, converged) identified initial conditions, for the three variables, at various times: initial time, end of the assimilation period, and end of the prediction period.

identified by the BFN becomes better near time step number 850 (whereas the end of the assimilation period corresponds to 720). Even if the quality of the velocity starts degrading quite quickly in the middle of the forecast period, the solution at the end is still better than the 4D-VAR one. The BFN height is also closer to the true solution than the 4D-VAR one, as shown in table 2 or on figure 6.

If for instance we stop the 4D-VAR after only 5 iterations, in order to have a similar computing time with the converged BFN, the 4D-VAR results are not as good, as shown in table 2. In this case, only the height at the initial time is better identified, and at the end of the assimilation period, the error is larger than for the BFN.

The fact that the BFN error decreases in time during the assimilation period has already been observed in [5]. This phenomenon is explained by the fact that the identified initial condition comes from a backward integration. During this integration, the backward model has reduced the error along its stable modes, but the stable modes of the backward model are exactly the unstable modes of the forward model. This means that the identified initial condition probably has a very small error along the unstable modes of the direct model. A direct integration of the model will then reduce the error along the stable modes, as there is nearly no error along the unstable modes.

3.4 Preprocessing of the 4D-VAR with the BFN algorithm

As the BFN algorithm (like all variational data assimilation schemes) estimates the initial condition at every iteration, it is possible to consider a coupled algorithm, in which the very first iterations are performed with the BFN, and then the next iterations with the 4D-VAR. As the BFN algorithm converges very quickly (in 5 iterations or less in all the experiments), we will consider a preprocessing with only 2 iterations of BFN, and then standard 4D-VAR iterations. We will study if the 4D-VAR convergence is reached more quickly, and if the identification is improved after only 3 iterations of 4D-VAR (in order to have a global cost of 5 iterations).

Table 3 shows the relative errors (as in the previous experiments) corresponding to this hybrid method. In 5 iterations, the results are much better than for the standard 4D-VAR (without any preprocessing). At convergence,

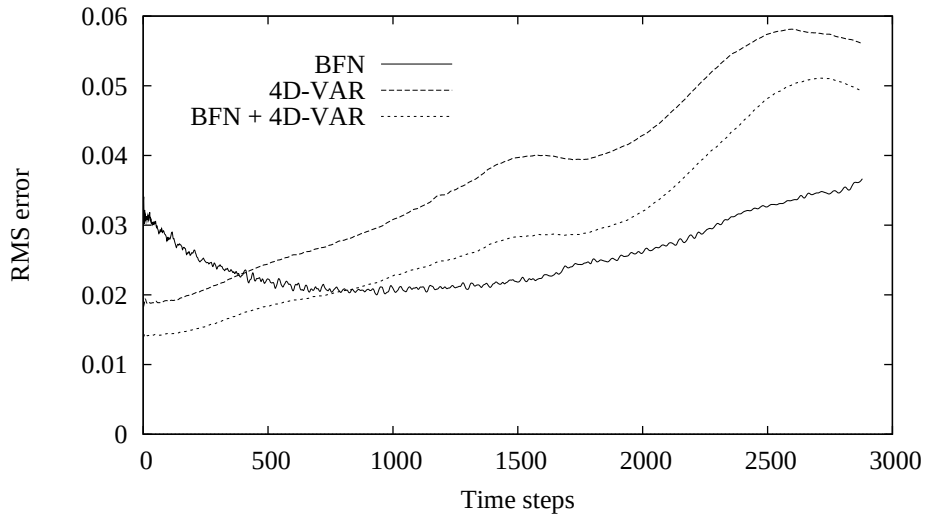

Figure 7: Relative difference between the true solution and the forecast trajectory corresponding to the BFN, 4D-VAR and BFN-preprocessed 4D-VAR identified initial conditions, after 5 iterations, versus time, for the height variable in the case of noisy observations.

the results are similar to what has been obtained without the BFN preprocessing. But in this case, the improvement concerns the number of iterations, as only 6 4D-VAR iterations are necessary to reach convergence, plus 2 preprocessing iterations of BFN, instead of 16 previously. The computing time needed to achieve convergence has been divided by 2, for almost the same results. And for 5 iterations only, the results obtained here correspond to what would have been obtained in 11 or 12 iterations of the standard 4D-VAR. This result is extremely noticeable, as it means that 2 iterations of BFN can save at least 5 iterations of 4D-VAR, or it can divide by 2 the computing time.

Figure 7 allows one to compare all these algorithms for the height variable. These results correspond to 5 iterations of BFN, 5 iterations of 4D-VAR, and 2 iterations of BFN followed by 3 iterations of 4D-VAR. We can see that during the assimilation period, the best solution is provided by the hybrid algorithm. Note that it would have been improved a little bit by the 4D-VAR at convergence, but we stop all algorithms after 5 iterations here. But during the forecast period, it is still provided by the BFN algorithm, except at the beginning. The results are comparable for u and v, except that the BFN gives a better solution a little bit earlier. In all the experiments performed, with several levels of noise and different background states, the hybrid method was the best during the assimilation period, and sometimes it was also the best at the beginning of the prediction period, but in most cases, the BFN solution is the best at the end. We can deduce from these results that a very cheap and simple way to largely improve the 4D-VAR algorithm is to perform a very few (2 or 3) iterations of BFN before starting the 4D-VAR minimization process.

3.5 Sensitivity of the BFN and 4D-VAR with respect to the time and space distribution of the observations

We now study the impact of the time and space distribution of the observations on the BFN algorithm. As the observations stabilize the backward integrations, it is important to see if a weak space or time distribution of the observations will degrade notably the performances or not. We still consider the noisy observations, as in the previous subsection. Recall that the standard distribution (previously used) is every 5 gridpoints in space (in each direction) and every 24 time steps. This distribution provides 8959 observations during the assimilation period (720 time steps), and the size of the control space (corresponding to the initial condition) is 19683. From the results obtained in the previous paragraphs, we will consider three different algorithms, stopped after 5 iterations: the BFN algorithm, the 4D-VAR algorithm, and the hybrid BFN 4D-VAR method (with 2 iterations of BFN and then 3 iterations of 4D-VAR). In all the experiments we have performed in this situation, the BFN reached convergence in at most 5 iterations, the 4D-VAR needed between 10 and 20 iterations to converge, and the hybrid method was close to convergence in 5 iterations (it needed between 7 and 12 iterations to converge, but the solution after 5 iterations was close to the solution at convergence).

Table 4 gives the relative errors between the true solution and the identified solution, for the three algorithms (BFN, 4D-VAR and hybrid method) at the initial time, the end of the assimilation period, and the end of the prediction period. Several space distributions have been considered (from every gridpoint to every 20 gridpoints) as well as several time distributions (from every 6 time steps to every 72 time steps). The couple in the first column of the table is the space and time frequency of the observations, the first row corresponding to the standard values: every 5 gridpoints and every 24 time steps. The largest number of observations corresponds to the $(1;6)$ couple, which represents 6561 observations at every observation time, and hence 793881 observations. The smallest number of observations corresponds to the $(20;72)$ couple, with 25 observations at every observation time, and 275 observations all over the assimilation period. The ratio between the size of the control vector and the number of observations varies from $\frac{1}{40}$ to 71.

Several conclusions can be drawn from this table. First of all, the global behaviour of each algorithm is relatively independent of the data distribution: the 4D-VAR and hybrid errors still increase in time, whereas the BFN error has a smaller value at the end of the assimilation period. The hybrid method is also almost every time better than the 4D-VAR (not only for 5 iterations, but for any number of iterations), and the convergence is usually reached 5 to 10 iterations earlier than for the 4D-VAR algorithm. Concerning the BFN algorithm, it is interesting to see that for a small amount of observations, the results are not much worse than for a standard data distribution. But on the contrary, if the number of observations increases (either with a better distribution in space or in time), the results are really improved. In the best case, the solution is identified with less than 0.5% error for the height and around 2% for the velocity. Recall that the observations have 20 to 40% errors.

In some sense, the BFN algorithm is quite insensitive to a (relatively) small number of observations, but on the other hand, it is extremely powerful with many observations. The 4D-VAR is more uniformly sensitive to the number of

		$t=0$			$t=T$			$t=4T$		
		h	u	v	h	u	v	h	u	v
	BFN	3.34%	9.51%	13.6%	2.15%	6.74%	10.1%	3.67%	17.0%	24.1%
(5;24)	4D-VAR	1.83%	11.3%	16.4%	2.68%	13.5%	18.9%	5.60%	25.9%	30.9%
	Hybrid	1.39%	7.34%	10.4%	1.98%	9.22%	12.8%	4.92%	22.2%	25.8%
	BFN	0.94%	2.90%	4.03%	0.72%	2.67%	3.69%	1.69%	8.48%	11.8%
(5;6)	4D-VAR	1.57%	6.12%	9.91%	2.02%	11.4%	13.8%	5.11%	22.3%	29.5%
	Hybrid	1.26%	4.94%	7.77%	1.88%	8.30%	11.4%	4.67%	19.5%	26.4%
	BFN	3.27%	9.31%	13.4%	2.22%	6.61%	10.4%	3.89%	16.5%	26.0%
(5;72)	4D-VAR	2.26%	13.1%	18.7%	3.58%	16.8%	24.1%	7.67%	31.2%	42.3%
	Hybrid	1.87%	7.72%	10.1%	2.58%	10.8%	14.0%	6.67%	23.1%	31.4%
	BFN	3.51%	9.39%	13.0%	2.26%	6.90%	10.2%	3.96%	17.7%	22.6%
(20;24)	4D-VAR	2.13%	14.2%	19.3%	3.21%	18.2%	21.7%	7.25%	25.8%	32.1%
	Hybrid	1.61%	6.94%	10.61%	2.39%	10.4%	13.7%	6.21%	23.9%	27.2%
	BFN	3.62%	9.78%	12.3%	2.37%	6.67%	10.2%	4.05%	18.3%	24.5%
(20;72)	4D-VAR	3.02%	13.7%	16.4%	4.15%	17.2%	21.3%	9.54%	18.4%	24.8%
	Hybrid	2.15%	6.41%	9.85%	2.95%	10.3%	14.2%	7.2%	24.2%	25.8%
	BFN	1.04%	3.52%	5.05%	0.71%	2.84%	4.09%	1.86%	8.54%	10.9%
(1;24)	4D-VAR	1.41%	4.31%	5.04%	1.74%	7.34%	9.12%	4.01%	12.5%	15.2%
	Hybrid	1.18%	2.81%	3.89%	1.52%	4.17%	7.21%	3.67%	10.5%	14.2%
	BFN	0.49%	1.91%	2.75%	0.40%	1.85%	2.57%	1.16%	5.56%	7.15%
(1;6)	4D-VAR	1.14%	3.51%	4.12%	1.62%	4.02%	4.52%	3.51%	9.75%	13.4%
	Hybrid	0.94%	2.40%	3.37%	1.36%	3.42%	4.98%	2.87%	7.21%	9.12%

Table 4: Relative error of the forecast solutions corresponding to the BFN, 4D-VAR and hybrid algorithms after 5 iterations, for the three variables, at different times, and for several spatio-temporal (every x gridpoints and every y time steps) distributions of observations.

	$t=0$			$t=T$			$t=4T$		
	h	u	v	h	u	v	h	u	v
BFN	3.38%	7.26%	11.9%	2.92%	7.07%	9.70%	9.40%	21.7%	32.0%
4D-VAR	2.27%	12.1%	15.4%	4.01%	13.6%	17.4%	14.6%	29.5%	40.9%
Hybrid	1.92%	6.83%	9.04%	3.21%	8.93%	12.4%	10.1%	24.5%	35.8%

Table 5: Relative error of the forecast solutions corresponding to the BFN, 4D-VAR and hybrid algorithms after 5 iterations, for the three variables, at different times, and using a different model for the assimilation.

observations, with a better identification of the velocity when the observations are more frequent in space.

3.6 Comparison between BFN and 4D-VAR with an imperfect model

We finally study the impact on the BFN algorithm of a model error. We assume that the noisy observations that have been extracted from a model trajectory, come from a real situation, and we will consider a different model for assimilating the data. We will consider different values of the model coefficients, in order to simulate different dynamics. The difference between the model used for extracting data and the model used for assimilating the data is a simple way to study the performance of a data assimilation scheme in a more realistic situation than twin experiments, in which the model is the same for both the generation of observations and assimilation process.

The coefficients of the assimilating model are the following:

$$r = 5.10^{-8}\ s^{-1}, \qquad \nu = 15\ m^2.s^{-1}, \qquad \tau_{max} = 0.015\ s^{-2}.$$

Table 5 gives the corresponding relative forecast errors for the three algorithms, using this different model coefficients for the assimilation, and also for the predictions. The errors on the initial condition are globally larger than before, and this is not surprising as the model used for assimilating the observations is not the same model that was used for their generation. But the BFN algorithm is less sensitive to this modification, as the error on the velocity is a little bit smaller than previously. The 4D-VAR and hybrid methods are both perturbed by the model modifications, but the hybrid scheme also identifies quite well the velocity at the initial time, like the BFN. Then, the errors decrease more or less at the beginning of the assimiltion period (more for the BFN than for the two other schemes), and then they increase, particularly during the forecast period. At time $t = 4T$, the errors are much larger than in the previous experiments. This is also not very surprising, as the model used for the predictions is not the same model as the one used for computing the reference trajectory.

From this experiment, we can see that the BFN scheme is a little bit less sensitive to model perturbations. By definition, the BFN scheme does not use the standard model equations, but different ones, with nudging terms. In some sense, the forward and backward feedback terms correct the model equations

with the observations, and the model equations are no more strong constraints, as in the 4D-VAR scheme. Like the other schemes with a weak model formulation (e.g. 4D-PSAS algorithm), the BFN algorithm is quite efficient in comparison with the 4D-VAR. Of course, we did not consider any model error term to be controlled by the 4D-VAR cost function, but we also did not change anything to the BFN scheme. For no additional cost, the BFN scheme is able to correct (at least partially) the model equations from the observations. As in the previous experiments, the use of a hybrid scheme, with a few BFN iterations before some 4D-VAR minimization iterations, improves notably the identification process. Either the minimization is stopped after a given number of iterations (like in many operational systems) and hence before convergence, and in this case, the identified initial condition and the forecasts are much better (15 to 40% smaller error), or the convergence of the algorithms is reached, and in this case, the hybrid scheme needs around 5 to 10 iterations (i.e. nearly half the number of iterations needed to achieve convergence) less than the 4D-VAR.

4 CONCLUSIONS

In the framework of synthetic data assimilation on a simple geophysical model, the back and forth nudging (BFN) algorithm appears extremely efficient. It combines the two main advantages of being extremely easy to implement (no linearization of the model equations, no adjoint state, no optimization algorithm) and extremely efficient in the very first iterations, as it can divide the estimation error on the initial condition by 10 in one or two iterations. Another interesting point is that the nudging term allows us to stabilize the backward numerical integration of the model equations, which are known to be ill-posed and unstable from a physical point of view. But the feedback term simultaneously regularizes and penalizes the backward model equations, and force the trajectory to stay close to the observations.

The BFN algorithm has been compared with the standard variational method (4D-VAR) in various experiments, and we studied the impact of observation noise, model error, and space/time distribution of the observations on these two algorithms. It has been shown that the BFN algorithm does not identify very well the initial condition, but the corresponding trajectory at the end of the assimilation trajectory is much closer to the true state. The 4D-VAR usually identifies a better initial state, but it is less efficient at the end of the assimilation period, and the BFN scheme provides better forecasts. It is also less sensitive to various perturbations on the observations or model.

All these results made us introduce a new hybrid scheme, in which a very small number of BFN iterations are performed (2 or 3 for instance), before providing the identified initial condition to the standard 4D-VAR algorithm. By doing this, the convergence of the 4D-VAR is reached more quickly, as it sometimes divides by two the number of iterations required. Also, for a fixed given number of iterations (or for a given computation time), the quality of the identified solution is significantly improved by this preprocessing (note that the number of 4D-VAR iterations is decreased by the number of BFN iterations in this scheme, in order to consider the same number of iterations in the standard 4D-VAR and the hybrid scheme).

Several improvements can still be made on the BFN scheme (and hence on the hybrid method). For example, the other model variables could also be controlled by the nudging term on the height, by considering for instance the geostrophic balance between the velocity and the space derivative of the height. Moreover, these results have now to be extended to more realistic situations, with a sophisticated ocean or atmosphere model and real satellite observations. Finally, the use of the background error covariances in the BFN scheme is currently investigated from a theoretical point, and may lead to additional improvements.

References

[1] Adcroft A, Marshall D. How slippery are piecewise-constant coastlines in numerical ocean models? *Tellus* 1998; **50**(1):95–108.

[2] Arakawa A, Lamb V. Computational design of the basic dynamical processes of the UCLA general circulation model. In *Methods in Computational Physics*, **17**:174–267, Academic Press, 1977.

[3] Asselin R. Frequency filter for time integrations. *Mon. Wea. Rev.* 1972; **100**:487–490.

[4] Auroux D, Blum J. Back and forth nudging algorithm for data assimilation problems. *C. R. Acad. Sci. Sér. I* 2005; **340**:873–878.

[5] Auroux D, Blum J. A nudging-based data assimilation method: the Back and Forth Nudging (BFN) algorithm. *Nonlin. Proc. Geophys.* 2008; **15**:305–319.

[6] Bennett AF. *Inverse Modeling of the Ocean and Atmosphere*. Cambridge University Press, Cambridge, 2002.

[7] Blayo E, Durbiano S, Vidard PA, Le Dimet FX. Reduced order strategies for variational data assimilation in oceanic models. In *Data assimilation for geophysical flows*, Springer-Verlag, 2003.

[8] Durbiano S. Vecteurs caractéristiques de modèles océaniques pour la réduction d'ordre en assimilation de données. PhD thesis, University of Grenoble, 2001.

[9] Evensen G, van Leeuwen PJ. An Ensemble Kalman Smoother for Nonlinear Dynamics. *Mon. Wea. Rev.* 1999; **128**:1852–1867.

[10] Gilbert JC, Lemaréchal C. Some numerical experiments with variable-storage quasi-Newton algorithms. *Math. Prog.* 1989; **45**:407–435.

[11] Hoke J, Anthes R. The initialization of numerical models by a dynamic-initialization techniques. *Month. Weather Rev.* 1976; **104**:1551–1556.

[12] Kalnay E. *Atmospheric modeling, data assimilation and predictability*, Cambridge University Press, 2003.

[13] Le Dimet FX, Talagrand O. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus* 1986; **38A**:97–110.

[14] Lorenc AC, Bell RS, Macpherson B. The Meteorological Office analysis correction data assimilation scheme. *Quart. J. Royal Meteor. Soc.* 1991; **117**:59–89.

[15] Lyne WH, Swinbank R, Birch NT. A data assimilation experiment and the global circulation during the FGGE special observing periods. *Quart. J. Roy. Meteor. Soc.* 1982; **108**:575–594.

[16] Verron J, Holland WR. Impacts de données d'altimétrie satellitaire sur les simulations numériques des circulations générales océaniques aux latitudes moyennes. *Annales Geophysicae* 1989; **7**(1):31–46.

[17] Verron J, Gourdeau L, Pham DT, Murtugudde R, Busalacchi AJ. An extended Kalman filter to assimilate satellite altimeter data into a nonlinear numerical model of the tropical Pacific Ocean: Method and validation. *J. Geophys. Res.* 1999; **104**:5441–5458.

[18] Vidard PA, Le Dimet FX, Piacentini A. Determination of optimal nudging coefficients. *Tellus* 2003; **55A**:1–15.

[19] Zou X, Navon IM, Le Dimet FX. An optimal nudging data assimilation scheme using parameter estimation. *Q. J. Roy. Meteorol. Soc.* 1992; **118**:1163–1186.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399